

1 What is claimed is:

2

3 1. A method of detecting edges in a compressed video sequence, the
4 compressed video sequence including at least one frame of block encoded video data, the
5 frame of block encoded video data including variable-length codes for transform
6 coefficients of blocks of pixels in the compressed video sequence, the transform
7 coefficients including a respective DC coefficient for each of the blocks of pixels, each
8 respective DC coefficient for at least some of the blocks of pixels being encoded as a
9 respective variable-length code having a length indicating a certain range of differences
10 in DC coefficient values between adjacent ones of the blocks of pixels, wherein the
11 method comprises:

12 decoding only the length of the respective variable-length code for the respective
13 DC coefficient for each of said at least some of the blocks of pixels in order to produce an
14 indication of whether or not the compressed video sequence includes an edge associated
15 with said each of said at least some of the blocks of pixels; and

16 performing a code length threshold comparison upon the length of the respective
17 variable-length code for the respective DC coefficient for said each of said at least some
18 of the blocks of pixels for producing at least one respective bit indicating whether or not
19 the compressed video sequence includes an edge associated with said each of said at least
20 some of the blocks of pixels.

21

22 2. The method as claimed in claim 1, wherein the compressed video
23 sequence is a color video sequence and there is a respective DC luminance coefficient or

1 a respective DC C_b chrominance coefficient or a respective DC C_r chrominance
2 coefficient for each of the blocks of pixels depending on a color channel of each of the
3 blocks of pixels, and the method includes:

4 decoding the length of the respective variable-length code for the respective DC
5 luminance coefficient or DC C_b chrominance coefficient or DC C_r chrominance
6 coefficient of said each of said at least some of the blocks of pixels; and

7 comparing the decoded length of the respective variable-length code for the
8 respective DC luminance coefficient or DC C_b chrominance coefficient or DC C_r
9 chrominance coefficient of said each of said at least some of the blocks of pixels to at
10 least one length threshold to produce at least one respective bit indicating whether or not
11 the compressed video sequence includes a luminance edge or a C_b chrominance edge or a
12 C_r chrominance edge associated with said each of said at least some of the blocks of
13 pixels.

14

15 3. The method as claimed in claim 1, wherein the compressed video
16 sequence is a color video sequence and there is a respective DC luminance coefficient or
17 a respective DC C_b chrominance coefficient or a respective DC C_r chrominance
18 coefficient for each of the blocks of pixels depending on a color channel of each of the
19 blocks of pixels, and the method includes:

20 decoding the length of the respective variable-length code for the respective DC
21 luminance coefficient of said each of said at least some of the blocks of pixels;

22 decoding the length of the respective variable-length code for the respective DC
23 C_b chrominance coefficient of said each of said at least some of the blocks of pixels;

1 decoding the length of the respective variable-length code for the respective DC
2 C_r chrominance coefficient of said each of said at least some of the blocks of pixels;
3 combining the length of the respective variable-length code for the respective DC
4 luminance coefficient of said each of said at least some of the blocks of pixels with the
5 lengths of the respective variable-length codes for the respective DC C_b and C_r
6 chrominance coefficients of said each of said at least some of the blocks of pixels to
7 produce a combined code length; and
8 wherein at least one code length threshold is compared to the combined code
9 length for producing at least one respective bit indicating whether or not the compressed
10 video sequence includes an edge associated with said each of said at least some of the
11 blocks of pixels.

12
13 4. The method as claimed in claim 3, wherein the combined code length is
14 produced by adding the length of the respective variable-length code for the respective
15 DC luminance coefficient of said each of said at least some of the blocks of pixels to the
16 sum of the lengths of the respective variable-length codes for the respective DC C_b and C_r
17 chrominance coefficients of said each of said at least some of the blocks of pixels.

18
19 5. The method as claimed in claim 1, which includes using a thinning filter
20 for filtering the respective bits indicating whether or not the compressed video sequence
21 includes an edge associated with each of said at least some of the blocks of pixels.

22

1 6. The method as claimed in claim 5, wherein the filtering of the respective
2 bits indicating whether or not the compressed video sequence includes an edge associated
3 with said each of said at least some of the blocks of pixels includes comparing the lengths
4 of the respective variable-length codes of the DC coefficients for adjacent blocks of
5 pixels in order to retain indications of edges associated with blocks of pixels having
6 longer variable-length codes for their respective DC coefficients and to exclude
7 indications of edges associated with blocks of pixels having shorter variable-length codes
8 for their respective DC coefficients.

9

10 7. The method as claimed in claim 6, wherein an indication of an edge
11 associated with a block of pixels having a shorter variable-length code of the respective
12 DC coefficients for a pair of adjacent blocks of pixels is not excluded upon comparing
13 signs of the respective DC coefficients for the pair of adjacent blocks of pixels and
14 finding that the signs are different.

15

16 8. The method as claimed in claim 1, which includes inspecting signs of the
17 respective DC coefficients for said at least some of the blocks of pixels, and based on the
18 signs of the respective DC coefficients for said at least some of the blocks of pixels and
19 based on prediction directions of the respective DC coefficients for said at least some of
20 the blocks of pixels and based on the respective bits indicating whether or not the
21 compressed video sequence includes an edge associated with said at least some of the
22 blocks of pixels, producing a first series of bits indicating whether or not the compressed
23 video sequence includes positive horizontal gradient component edges associated with

1 said at least some of the blocks of pixels, and producing a second series of bits indicating
2 whether or not the compressed video sequence includes negative horizontal gradient
3 component edges associated with said at least some of the blocks of pixels.

4

5 9. The method as claimed in claim 1, which includes inspecting signs of the
6 respective DC coefficients for said at least some of the blocks of pixels, and based on the
7 signs of the respective DC coefficients for said at least some of the blocks of pixels and
8 based on prediction directions of the respective DC coefficients for said at least some of
9 the blocks of pixels and based on the respective bits indicating whether or not the
10 compressed video sequence includes an edge associated with said at least some of the
11 blocks of pixels, producing a first series of bits indicating whether or not the compressed
12 video sequence includes positive vertical gradient component edges associated with said
13 at least some of the blocks of pixels, and producing a second series of bits indicating
14 whether or not the compressed video sequence includes negative vertical gradient
15 component edges associated with said at least some of the blocks of pixels.

16

17 10. The method as claimed in claim 1, wherein the transform coefficients
18 include respective horizontal frequency transform coefficients and respective vertical
19 frequency transform coefficients for each block of pixels, and the method includes
20 inspecting a lowest nonzero horizontal frequency transform coefficient and a lowest
21 nonzero vertical frequency transform coefficient for at least one of the blocks of pixels to
22 determine orientation of an edge associated with said at least one of the blocks of pixels.

23

1 11. The method as claimed in claim 1, wherein the transform coefficients
2 include respective horizontal frequency transform coefficients and respective vertical
3 frequency transform coefficients for each block of pixels, and the method includes using
4 a lowest nonzero horizontal frequency transform coefficient and a lowest nonzero vertical
5 frequency transform coefficient for at least one of the blocks of pixels for computing an
6 approximate gradient vector of an edge associated with said at least one of the blocks of
7 pixels.

8

9 12. A method of detecting edges in a compressed video sequence, the
10 compressed video sequence including at least one I-frame of MPEG video data, the I-
11 frame of MPEG video data including variable-length codes for DCT coefficients of 8x8
12 pixel blocks in the compressed video sequence, the DCT coefficients including a
13 respective DC coefficient for each of the 8x8 pixel blocks, each respective DC coefficient
14 for at least some of the 8x8 pixel blocks being encoded as a respective variable-length
15 code having a length indicating a certain range of differences in DC coefficient values
16 between adjacent ones of the 8x8 pixel blocks, wherein the method comprises:

17 decoding only the length of the respective variable-length code for the respective
18 DC coefficient for each of said at least some of the 8x8 pixel blocks in order to produce
19 an indication of whether or not the compressed video sequence includes an edge
20 associated with said each of said at least some of the 8x8 pixel blocks; and

21 performing a code length threshold comparison upon the length of the respective
22 variable-length code for the respective DC coefficient for said each of said at least some
23 of the 8x8 pixel blocks for producing at least one respective bit indicating whether or not

1 the compressed video sequence includes an edge associated with said each of said at least
2 some of the 8x8 pixel blocks.

3

4 13. The method as claimed in claim 12, wherein the compressed video
5 sequence is a color video sequence and there is a respective DC luminance coefficient or
6 a respective DC C_b chrominance coefficient or a respective DC C_r chrominance
7 coefficient for each of the 8x8 pixel blocks depending on a color channel of each of the
8 8x8 pixel blocks, and the method includes:

9 decoding the length of the respective variable-length code for the respective DC
10 luminance coefficient or DC C_b chrominance coefficient or DC C_r chrominance
11 coefficient of said each of said at least some of the 8x8 pixel blocks; and

12 comparing the decoded length of the respective variable-length code for the
13 respective DC luminance coefficient or DC C_b chrominance coefficient or DC C_r
14 chrominance coefficient of said each of said at least some 8x8 pixel blocks to at least one
15 length threshold to produce at least one respective bit indicating whether or not the
16 compressed video sequence includes a luminance edge or a C_b chrominance edge or a C_r
17 chrominance edge associated with said each of said at least some of the 8x8 pixel blocks.

18

19 14. The method as claimed in claim 12, wherein the compressed video
20 sequence is a color video sequence and there is a respective DC luminance coefficient or
21 a respective DC C_b chrominance coefficient or a respective DC C_r chrominance
22 coefficient for each of the 8x8 pixel blocks depending on a color channel of each of the
23 8x8 pixel blocks, and the method includes:

1 decoding the length of the respective variable-length code for the respective DC
2 luminance coefficient of said each of said at least some of the 8x8 pixel blocks;
3 decoding the length of the respective variable-length code for the respective DC
4 C_b chrominance coefficient of said each of said at least some of the 8x8 pixel blocks;
5 decoding the length of the respective variable-length code for the respective DC
6 C_r chrominance coefficient of said each of said at least some of the 8x8 pixel blocks;
7 combining the length of the respective variable-length code for the respective DC
8 luminance coefficient of said each of said at least some of the 8x8 pixel blocks with the
9 lengths of the respective variable-length codes for the respective DC C_b and C_r
10 chrominance coefficients of said each of said at least some of the 8x8 pixel blocks to
11 produce a combined code length; and
12 wherein at least one code length threshold is compared to the combined code
13 length for producing at least one respective bit indicating whether or not the compressed
14 video sequence includes an edge associated with said each of said at least some of the
15 8x8 pixel blocks.

16

17 15. The method as claimed in claim 14, wherein the combined code length is
18 produced by adding the length of the respective variable-length code for the respective
19 DC luminance coefficient of said each of said at least some of the 8x8 pixel blocks to the
20 sum of the lengths of the respective variable-length codes for the respective DC C_b and C_r
21 chrominance coefficients of said each of said at least some of the 8x8 pixel blocks.

22

1 16. The method as claimed in claim 12, which includes using a thinning filter
2 for filtering the respective bits indicating whether or not the compressed video sequence
3 includes an edge associated with each of said at least some of the 8x8 pixel blocks.

4

5 17. The method as claimed in claim 16, wherein the filtering of the respective
6 bits indicating whether or not the compressed video sequence includes an edge associated
7 with said each of said at least some of the 8x8 pixel blocks includes comparing the
8 lengths of the respective variable-length codes of the DC coefficients for adjacent 8x8
9 pixel blocks in order to retain indications of edges associated with 8x8 pixel blocks
10 having longer variable-length codes for their respective DC coefficients and to exclude
11 indications of edges associated with 8x8 pixel blocks having shorter variable-length
12 codes for their respective DC coefficients.

13

14 18. The method as claimed in claim 17, wherein an indication of an edge
15 associated with a block of pixels having a shorter variable-length code of the respective
16 DC coefficients for a pair of adjacent 8x8 pixel blocks is not excluded upon comparing
17 signs of the respective DC coefficients for the pair of adjacent 8x8 pixel blocks and
18 finding that the signs are different.

19

20 19. The method as claimed in claim 12, which includes inspecting signs of the
21 respective DC coefficients for said at least some of the 8x8 pixel blocks, and based on the
22 signs of the respective DC coefficients for said at least some of the 8x8 pixel blocks and
23 based on prediction directions of the respective DC coefficients for said at least some of

1 the 8x8 pixel blocks and based on the respective bits indicating whether or not the
2 compressed video sequence includes an edge associated with said at least some of the 8x8
3 pixel blocks, producing a first series of bits indicating whether or not the compressed
4 video sequence includes positive horizontal gradient component edges associated with
5 said at least some of the 8x8 pixel blocks, and producing a second series of bits indicating
6 whether or not the compressed video sequence includes negative horizontal gradient
7 component edges associated with said at least some of the 8x8 pixel blocks.

8 The method as claimed in claim 11, which includes inspecting signs of the
9 respective DC coefficients for said at least some of the 8x8 pixel blocks, and based on the
10 signs of the respective DC coefficients for said at least some of the 8x8 pixel blocks and
11 based on prediction directions of the respective DC coefficients for said at least some of
12 the 8x8 pixel blocks and based on the respective bits indicating whether or not the
13 compressed video sequence includes an edge associated with said at least some of the 8x8
14 pixel blocks, producing a first series of bits indicating whether or not the compressed
15 video sequence includes positive vertical gradient component edges associated with said
16 at least some of the 8x8 pixel blocks, and producing a second series of bits indicating
17 whether or not the compressed video sequence includes negative vertical gradient
18 component edges associated with said at least some of the 8x8 pixel blocks.

19
20 20. The method as claimed in claim 12, wherein the DCT coefficients include
21 respective horizontal frequency DCT coefficients and respective vertical frequency DCT
22 coefficients for each of the 8x8 pixel blocks, and the method includes inspecting a lowest
23 nonzero horizontal frequency DCT coefficient and a lowest nonzero vertical frequency

1 DCT coefficient for at least one of the 8x8 pixel blocks to determine orientation of an
2 edge associated with said at least one of the 8x8 pixel blocks.

3

4 21. The method as claimed in claim 12, wherein the DCT coefficients include
5 respective horizontal frequency DCT coefficients and respective vertical frequency DCT
6 coefficients for each of the 8x8 pixel blocks, and the method includes using a lowest
7 nonzero horizontal frequency DCT coefficient and a lowest nonzero vertical frequency
8 DCT coefficient for at least one of the 8x8 pixel blocks for computing an approximate
9 gradient vector of an edge associated with said at least one of the 8x8 pixel blocks.

10

11 22. A method of detecting a scene change between I-frames of MPEG video
12 data, said method comprising:

13 a) detecting edges in images represented by the I-frames by decoding lengths of
14 variable-length codes for DCT DC coefficients of 8x8 pixel blocks in the I-frames and
15 performing code length threshold comparisons upon the decoded code lengths to produce
16 respective edge indications for each of the I-frames; and

17 b) comparing the edge indications between the I-frames in order to signal a scene
18 change when there is a significant change in the edge indications between the I-frames.

19

20 23. The method as claimed in claim 22, wherein the detecting of edges in the
21 images includes producing a frame of bits for at least one of the I-frames, the frame of
22 bits including at least one respective bit for each of the 8x8 pixel blocks in said at least

1 one of the I-frames, and storing the frame of bits for said at least one of the I-frames in a
2 frame buffer, and

3 wherein the comparing of the edge indications between the I-frames includes
4 accessing the frame of bits in the frame buffer for comparing the edge indications for an
5 I-frame following said at least one of the I-frames to the edge indications for said at least
6 one of the I-frames.

7

8 24. The method as claimed in claim 22, wherein the comparing of the edge
9 indications between the I-frames includes extracting features from the edge indications
10 for each of the I-frames and comparing features extracted between the I-frames.

11

12 25. The method as claimed in claim 22, wherein the comparing of the edge
13 indications between the I-frames includes computing characteristics of the edge
14 indications for each of the I-frames and comparing the characteristics of the edge
15 indications between the I-frames.

16

17 26. The method as claimed in claim 22, wherein the 8x8 pixel blocks each
18 have a respective DC luminance coefficient or a respective DC C_b chrominance
19 coefficient or a respective DC C_r chrominance coefficient depending on a color channel
20 of the 8x8 pixel block, and the method includes:

21 decoding the length of the respective variable-length code for the respective DC
22 luminance coefficient of each of at least some of the 8x8 pixel blocks;

1 decoding the length of the respective variable-length code for the respective DC
2 C_b chrominance coefficient of said each of said at least some of the 8x8 pixel blocks;
3 decoding the length of the respective variable-length code for the respective DC
4 C_r chrominance coefficient of said each of said at least some of the 8x8 pixel blocks;
5 combining the length of the respective variable-length code for the respective DC
6 luminance coefficient of said each of said at least some of the 8x8 pixel blocks with the
7 lengths of the respective variable-length codes for the respective DC C_b and C_r
8 chrominance coefficients of said each of said at least some of the 8x8 pixel blocks to
9 produce a combined code length; and
10 comparing at least one code length threshold to the combined code length for
11 producing at least one respective bit providing an edge indication for said each of said at
12 least some of the 8x8 pixel blocks.

13
14 27. The method as claimed in claim 26, wherein the combined code length is
15 produced by adding the length of the respective variable-length code for the respective
16 DC luminance coefficient of said each of said at least some of the 8x8 pixel blocks to the
17 sum of the lengths of the respective variable-length codes for the respective DC C_b and C_r
18 chrominance coefficients of said each of said at least some of the 8x8 pixel blocks.

19
20 28. The method as claimed in claim 22, which includes using a thinning filter
21 for filtering the respective edge indications.

22

1 29. The method as claimed in claim 28, wherein the filtering of the respective
2 edge indications includes comparing the lengths of the respective variable-length codes
3 of the DC coefficients for adjacent 8x8 pixel blocks in order to retain indications of edges
4 associated with 8x8 pixel blocks having longer variable-length codes for their respective
5 DC coefficients and to exclude indications of edges associated with 8x8 pixel blocks
6 having shorter variable-length codes for their respective DC coefficients.

7

8 30. The method as claimed in claim 29, wherein an indication of an edge
9 associated with a block of pixels having a shorter variable-length code of the respective
10 DC coefficients for a pair of adjacent 8x8 pixel blocks is not excluded upon comparing
11 signs of the respective DC coefficients for the pair of adjacent 8x8 pixel blocks and
12 finding that the signs are different.

13

14 31. The method as claimed in claim 22, which includes inspecting signs of the
15 respective DC coefficients for at least some of the 8x8 pixel blocks, and based on the
16 signs of the respective DC coefficients for said at least some of the 8x8 pixel blocks and
17 based on prediction directions of the respective DC coefficients for said at least some of
18 the 8x8 pixel blocks and based on respective bits indicating whether or not an edge is
19 associated with said at least some of the 8x8 pixel blocks, producing a first series of bits
20 indicating whether or not positive horizontal gradient component edges are associated
21 with said at least some of the 8x8 pixel blocks, and producing a second series of bits
22 indicating whether or not negative horizontal gradient component edges are associated
23 with said at least some of the 8x8 pixel blocks.

1

2 32. The method as claimed in claim 22, which includes inspecting signs of the
3 respective DC coefficients for at least some of the 8x8 pixel blocks, and based on the
4 signs of the respective DC coefficients for said at least some of the 8x8 pixel blocks and
5 based on prediction directions of the respective DC coefficients for said at least some of
6 the 8x8 pixel blocks and based on respective bits indicating whether or not an edge is
7 associated with said at least some of the 8x8 pixel blocks, producing a first series of bits
8 indicating whether or not positive vertical gradient component edges are associated with
9 said at least some of the 8x8 pixel blocks, and producing a second series of bits indicating
10 whether or not negative vertical gradient component edges are associated with said at
11 least some of the 8x8 pixel blocks.

12

13 33. The method as claimed in claim 22, wherein the DCT coefficients include
14 respective horizontal frequency DCT coefficients and respective vertical frequency DCT
15 coefficients for each of the 8x8 pixel blocks, and the method includes inspecting a lowest
16 nonzero horizontal frequency DCT coefficient and a lowest nonzero vertical frequency
17 DCT coefficient for at least one of the 8x8 pixel blocks to determine orientation of an
18 edge associated with said at least one of the 8x8 pixel blocks.

19

20 34. The method as claimed in claim 22, wherein the DCT coefficients include
21 respective horizontal frequency DCT coefficients and respective vertical frequency DCT
22 coefficients for each of the 8x8 pixel blocks, and the method includes using a lowest
23 nonzero horizontal frequency DCT coefficient and a lowest nonzero vertical frequency

1 DCT coefficient for at least one of the 8x8 pixel blocks for computing an approximate
2 gradient vector of an edge associated with said at least one of the 8x8 pixel blocks.

3

4 35. A method of detecting a scene change between I-frames of MPEG video
5 data, each of the I-frames including a series of 8x8 pixel blocks, said method comprising:

6 a) detecting edges in images represented by the I-frames to produce a series of
7 respective bits indicating whether or not an edge is associated with at least some of the
8 8x8 pixel blocks;

9 b) filtering the series of the respective bits indicating whether or not an edge is
10 associated with said at least some of the 8x8 pixel blocks with a thinning filter in order to
11 produce a filtered series of respective bits including more significant edge indications and
12 excluding less significant edge indications; and

13 c) operating a digital processor to process the filtered series of respective bits in
14 order to signal a scene change when there is a significant change in features between the
15 I-frames.

16

17 36. The method as claimed in claim 35, which includes storing a frame of bits
18 of the filtered series of respective bits for at least one of the I-frames in a frame buffer,
19 and wherein the digital processor accesses the frame of bits in the frame buffer for
20 comparing edge indications for an I-frame following said at least one of the I-frames to
21 edge indications for said at least one of the I-frames.

22

1 37. The method as claimed in claim 35, wherein the processor extracts
2 features from the filtered series of respective bits for each of the I-frames and compares
3 features extracted from at least one of the I-frames to features extracted from an I-frame
4 following said at least one of the I-frames.

5

6 38. The method as claimed in claim 35, wherein the processor computes
7 characteristics of the filtered series of respective bits for each of the I-frames and
8 compares the characteristics of the filtered series of respective bits for at least one of the
9 I-frames to the characteristics of the filtered series of respective bits for an I-frame
10 following said at least one of the I-frames.

11

12 39. The method as claimed in claim 35, wherein the detecting of edges in
13 images represented by the I-frames includes decoding lengths of variable-length codes of
14 DCT DC transform coefficients of each of said at least some of the 8x8 pixel blocks and
15 performing a length threshold comparison upon the decoded lengths of the variable-
16 length codes to produce the series of respective bits indicating whether or not an edge is
17 associated with said at least some of the 8x8 pixel blocks.

18

19 40. The method as claimed in claim 39, wherein the 8x8 pixel blocks each
20 have a respective DC luminance coefficient or a respective DC C_b chrominance
21 coefficient or a respective DC C_r chrominance coefficient depending on a color channel
22 of the 8x8 pixel block, and the method includes:

1 decoding the length of the respective variable-length code for the respective DC
2 luminance coefficient of said each of said at least some of the 8x8 pixel blocks;
3 decoding the length of the respective variable-length code for the respective DC
4 C_b chrominance coefficient of said each of said at least some of the 8x8 pixel blocks;
5 decoding the length of the respective variable-length code for the respective DC
6 C_r chrominance coefficient of said each of said at least some of the 8x8 pixel blocks;
7 combining the length of the respective variable-length code for the respective DC
8 luminance coefficient of said each of said at least some of the 8x8 pixel blocks with the
9 lengths of the respective variable-length codes for the respective DC C_b and C_r
10 chrominance coefficients of said each of said at least some of the 8x8 pixel blocks to
11 produce a combined code length; and
12 comparing at least one code length threshold to the combined code length for
13 producing at least one respective bit providing an edge indication for said each of said at
14 least some of the 8x8 pixel blocks.

15
16 41. The method as claimed in claim 40, wherein the combined code length is
17 produced by adding the length of the respective variable-length code for the respective
18 DC luminance coefficient of said each of said at least some of the 8x8 pixel blocks to the
19 sum of the lengths of the respective variable-length codes for the respective DC C_b and C_r
20 chrominance coefficients of said each of said at least some of the 8x8 pixel blocks.

21
22 42. The method as claimed in claim 39, wherein the filtering of the series of
23 the respective bits indicating whether or not an edge is associated with said at least some

1 of the 8x8 pixel blocks includes comparing the decoded lengths of the variable-length
2 codes for adjacent 8x8 pixel blocks in order to retain edge indications for 8x8 pixel
3 blocks having longer decoded code lengths and to exclude edge indications for 8x8 pixel
4 blocks having shorter decoded code lengths.

5

6 43. The method as claimed in claim 42, wherein an indication of an edge
7 associated with an 8x8 pixel block having a shorter variable-length code of the respective
8 DC coefficients for a pair of adjacent 8x8 pixel blocks is not excluded upon comparing
9 signs of the respective DC coefficients for the pair of adjacent 8x8 pixel blocks and
10 finding that the signs are different.

11

12 44. The method as claimed in claim 35, which includes inspecting signs of the
13 respective DC coefficients for said at least some of the 8x8 pixel blocks, and based on the
14 signs of the respective DC coefficients for said at least some of the 8x8 pixel blocks and
15 based on prediction directions of the respective DC coefficients for said at least some of
16 the 8x8 pixel blocks and based on the respective bits indicating whether or not an edge is
17 associated with said at least some of the 8x8 pixel blocks, producing a first series of bits
18 indicating whether or not positive horizontal gradient component edges are associated
19 with said at least some of the 8x8 pixel blocks, and producing a second series of bits
20 indicating whether or not negative horizontal gradient component edges are associated
21 with said at least some of the 8x8 pixel blocks.

22

1 45. The method as claimed in claim 35, which includes inspecting signs of the
2 respective DC coefficients for said at least some of the 8x8 pixel blocks, and based on the
3 signs of the respective DC coefficients for said at least some of the 8x8 pixel blocks and
4 based on prediction directions of the respective DC coefficients for said at least some of
5 the 8x8 pixel blocks and based on the respective bits indicating whether or not an edge is
6 associated with said at least some of the 8x8 pixel blocks, producing a first series of bits
7 indicating whether or not positive vertical gradient component edges are associated with
8 said at least some of the 8x8 pixel blocks, and producing a second series of bits indicating
9 whether or not negative vertical gradient component edges are associated with said at
10 least some of the 8x8 pixel blocks.

11

12 46. The method as claimed in claim 35, which includes inspecting a lowest
13 nonzero horizontal frequency DCT coefficient and a lowest nonzero vertical frequency
14 DCT coefficient for at least one of the 8x8 pixel blocks to determine orientation of an
15 edge associated with said at least one of the 8x8 pixel blocks.

16

17 47. The method as claimed in claim 35, which includes using a lowest
18 nonzero horizontal frequency DCT coefficient and a lowest nonzero vertical frequency
19 DCT coefficient for at least one of the 8x8 pixel blocks for computing an approximate
20 gradient vector of an edge associated with said at least one of the 8x8 pixel blocks.

21